



Sesión tutorial introductoria sobre requisitos y trabajo en equipo

Sesión Técnica de Calidad de Software

12 de noviembre de 2008

Luis Fernández Sanz

Universidad de Alcalá

www.ati.es/gtcalidadsoft

- 1 -

Definiciones

- Requisito (IEEE std. 610):
 - Condición o capacidad
 - Necesitada por un usuario para resolver un problema o lograr un objetivo
 - Que debe cumplir o poseer un sistema para satisfacer un contrato, estándar, especificación u otro documento impuesto formalmente
- Análisis de requisitos (IEEE std. 610):
 - Proceso de estudio de las necesidades del usuario para llegar a una definición de requisitos del software o sistema
 - Proceso de estudiar y refinar los requisitos del software o sistema

- 2 -

Requisitos funcionales y no funcionales

- Requisito funcional (IEEE std. 610):
 - **Requisito que especifica una función que un sistema o componente debe ser capaz de realizar**
 - **Ejemplos de un sistema de gestión de pedidos:**
 - Obtener estadísticas pedidos
 - Introducir datos de pedido
- Requisito no funcional:
 - **Requisito que especifica una característica de funcionamiento**
 - Características de ejecución de las funciones
 - Seguridad, rendimiento, facilidad de uso, capacidad, etc.
 - **Ejemplos de un sistema de gestión de notas:**
 - Capaz de gestionar 1000 pedidos diarios y con hasta 100 referencias
 - Seguridad de acceso para clientes basada en clave de 128 bits

- 3 -

Proceso de análisis

Fases:

1. Identificar las fuentes de información
2. Extraer información (*elicitation*):
 - Preguntas apropiadas, etc.
3. Analizar la información:
 - Coherencia, completión, entender, etc.
4. Confirmar con los usuarios
5. Sintetizar los requisitos en un documento
 - Que sirva de referencia para el desarrollo



- 4 -

Filosofía del proceso

- El analista se constituye como nexo entre cliente y equipo de desarrollo
- El proceso de análisis:
 - Esfuerzo intensivo de ordenación de la información
 - Análisis: mínimo del 15% del proyecto
- El analista: mejor conocedor de la problemática y el dominio del sistema
- El analista asume el rol (a veces ingrato) de determinar y documentar los requisitos
- El analista documenta los requisitos para que el resto del equipo:
 - No tenga que pensar en ellos ni repetir el análisis
 - Sólo consulte y entienda fácilmente los objetivos del sistema

- 5 -

Recogida de información

- Entrevistas
- Desarrollo conjunto de aplicaciones (JAD)
- ETHICS
- Prototipado
- Observación
- Cuestionarios
- Estudio de documentación
- Tormenta de ideas (brainstorming)



- 7 -

Especificación de requisitos del software

- Especificación de Requisitos del Software (ERS)
 - **documentación de los requisitos esenciales (funciones, rendimiento, diseño, restricciones y atributos) del software y sus interfaces externas**
- Características:
 - **Información veraz**
 - **Comunicación eficaz**
- ERS debe definir el qué, no el cómo, el cuándo
 - **Describir todos los requisitos pero no más**
 - **No describir detalles de diseño, proyecto, verificación**

- 8 -

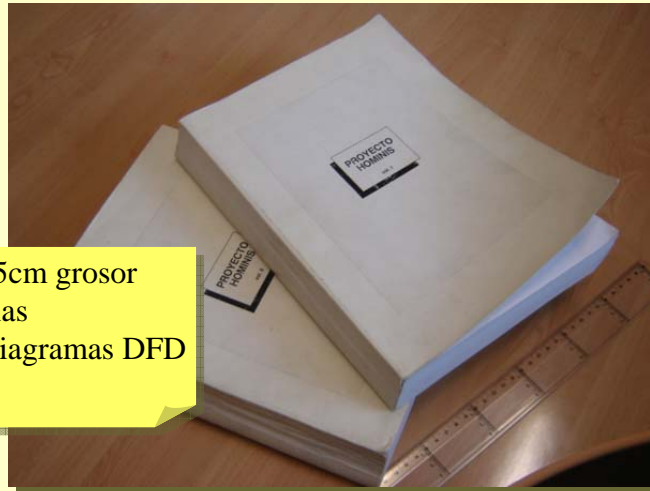
Una buena ERS

- IEEE Std. 830:
 - No ambigua
 - Completa
 - Fácil de verificar
 - Coherente/consistente
 - Fácil de modificar
 - Ordenación por prioridades y/o estabilidad
 - Facilidad para identificar fuente y efectos de cada requisito (trazabilidad)
 - Facilidad de uso en explotación y mantenimiento



- 9 -

El documento



2 tomos, 5cm grosor
570 páginas
Texto + diagramas DFD
Peso: ¿?

- 10 -

IEEE std. 830

- Introducción
 - **Objetivo, ámbito, definiciones y siglas, referencias**
- Descripción general
 - **Visión general de producto, funciones, usuarios, limitaciones generales, supuestos y dependencias**
- Requisitos específicos numerados
 - **Funcionales (Entrada, Proceso, Salida)**
 - **Interfaz**
 - **Rendimiento, restricciones de diseño, calidad, otros**
- Apéndice
- Tabla

- 11 -

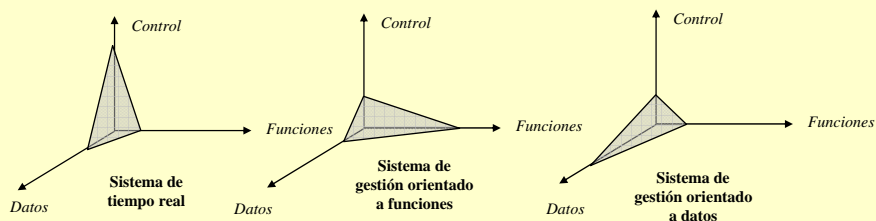
Técnicas de especificación

- Formales:
 - Basadas en lógica formal: ej., notación Z
 - $usados \cap libres = \emptyset$
- Semiformales:
 - Gráficas: notación definida
 - DFD (con otras técnicas), etc.
 - Diagramas de actividad, etc.
 - Textual: gramática semiformal
 - Pseudocódigo de proceso, BNF
 - Casos de uso
 - Plantilla: formulario
 - Entrada del DD (DDE), fichas, etc.
 - Matriciales: para comprobar más que definir
 - Matriz entidad/función, requisitos/casos de uso, etc.

- 12 -

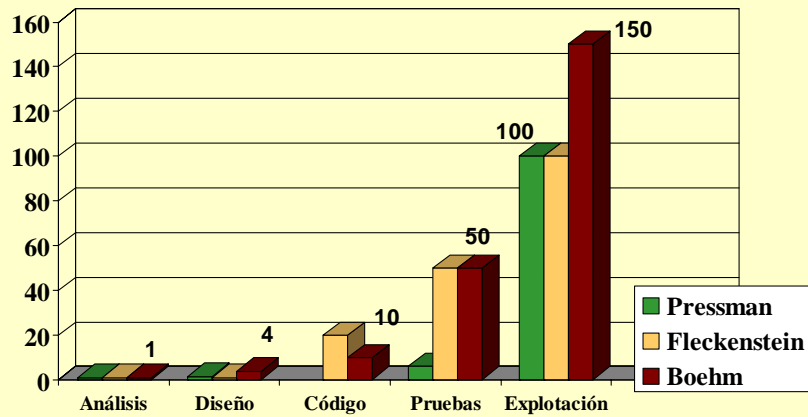
Dimensiones de especificación

- Tres perspectivas:
 - **Funciones:** qué hace
 - **Datos:** qué información
 - **Tiempo/eventos/control:** cuándo sucede algo
- Tres dimensiones:



- 13 -

Razones económicas



- 14 -

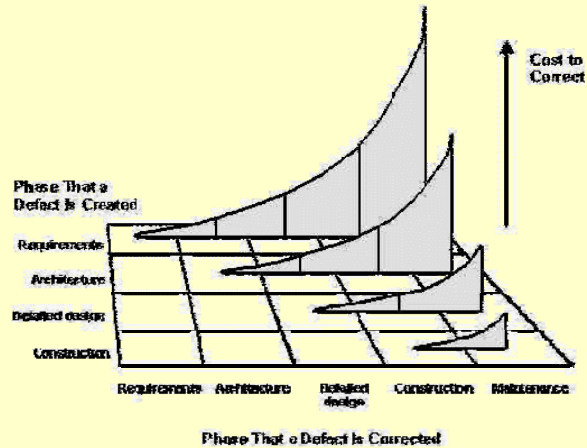
Prevención vs reacción

Coste	Def/KLOC acum.		Coste	Def/KLOC acum.
0	20	Requisitos 20 defectos	10	10
0	40	Diseño 20 defectos	25	15
0	60	Codificación 20 defectos	42	18
480	12	Pruebas reducción del 80%	182	4
1.680	0	Explotación 20 defectos	582	0
<u>QA Tradicional</u>			<u>QA Integral</u>	

- 15 -

Costes incrementales

60% de total de defectos existen ya en la fase de diseño (Gilb, 1988)



- 16 -

¿Sólo técnica?

- RENTIC:
 - Serie de estudios de requisitos de empleo desde 1998: análisis de 2997 ofertas
 - 239 de analistas

Informática en general	%
Trabajo en equipo	10,88%
Dinámico/Proactivo/iniciativa	9,62%

	Analistas	%
Orientación a cliente	Trabajo en equipo	10,44%
Dotes de comunicación	Dinámico/Proactivo/iniciativa	7,41%
Autonomía/independencia	Orientación a cliente	6,01%
Capacidad de organización	Habilidades comunicativas	4,10%
Flexibilidad	Orientado a resultados	3,44%
	Capacidad analítica	3,37%
	Capacidad de organización/gestión	3,37%

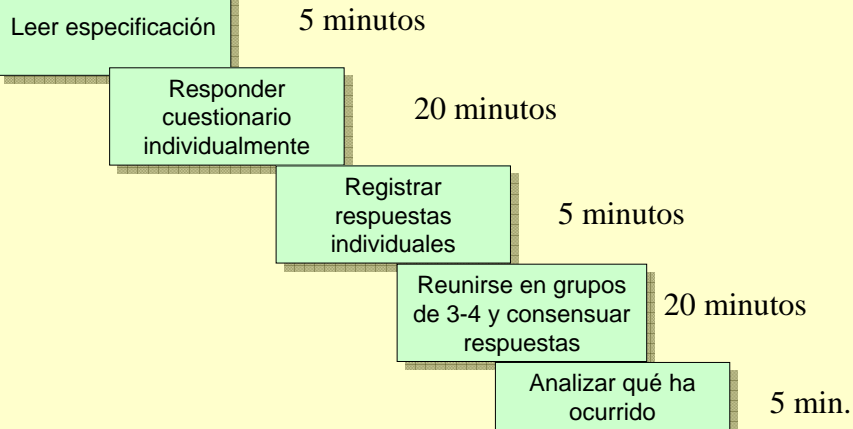
- 17 -

Teamwork

- El trabajo en equipo:
 - ¿Genera ventajas (sinergias)?: ¿ $1+1 = 3$?
 - ¿Cuál es nuestra experiencia?
 - ¿Qué beneficios genera realmente?
 - ¿Mejores resultados?
 - ¿Más comodidad, satisfacción, etc.?
 - ¿Permite ayudar en aspectos adicionales en el análisis de requisitos?

- 18 -

Especificaciones y trabajo en equipo



- 19 -

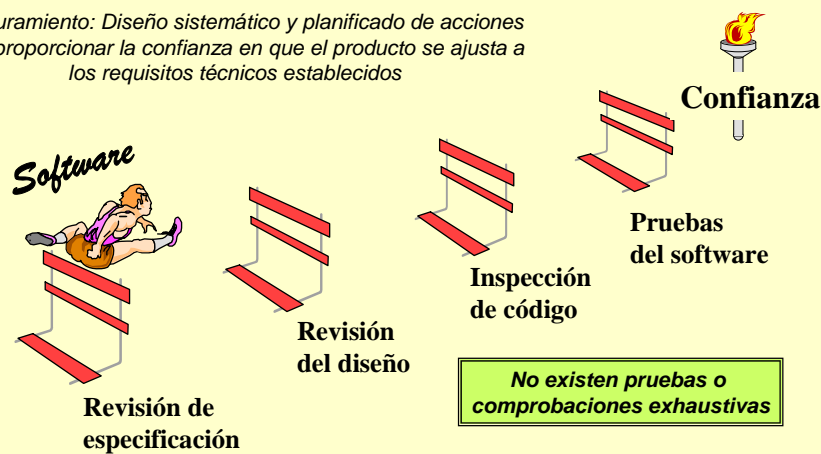
Requisitos

- El elemento clave para el desarrollo
- Beizer: 47% defectos por problemas en requisitos
- Tipología:
 - Incompletos (73,4%), erróneos, ausentes (11,2%), generalización (4,6%)
 - Compleción (37,5%), presentación (34,7%), cambios (11,2%), incorrectos (8,7%)
- Actuar con mejores prácticas y con mejores controles
- Son la base del desarrollo, de las pruebas, de la gestión de proyecto, etc.

- 22 -

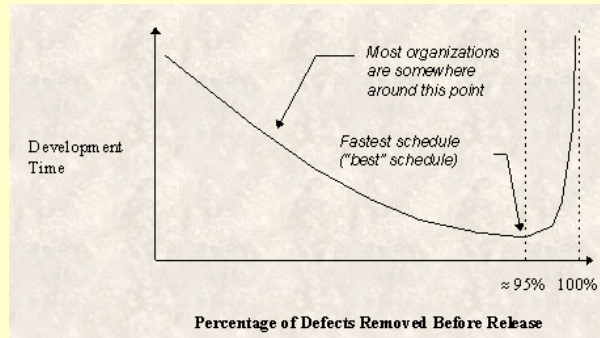
QA integral

Aseguramiento: Diseño sistemático y planificado de acciones para proporcionar la confianza en que el producto se ajusta a los requisitos técnicos establecidos



- 23 -

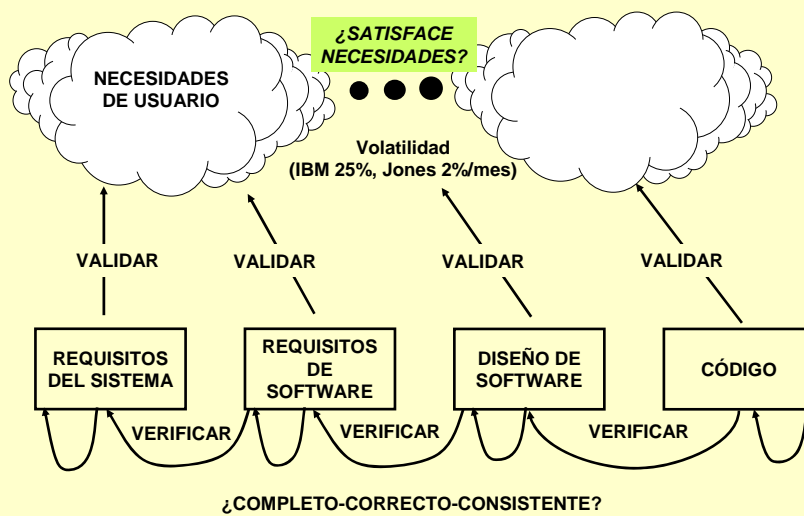
Equilibrio



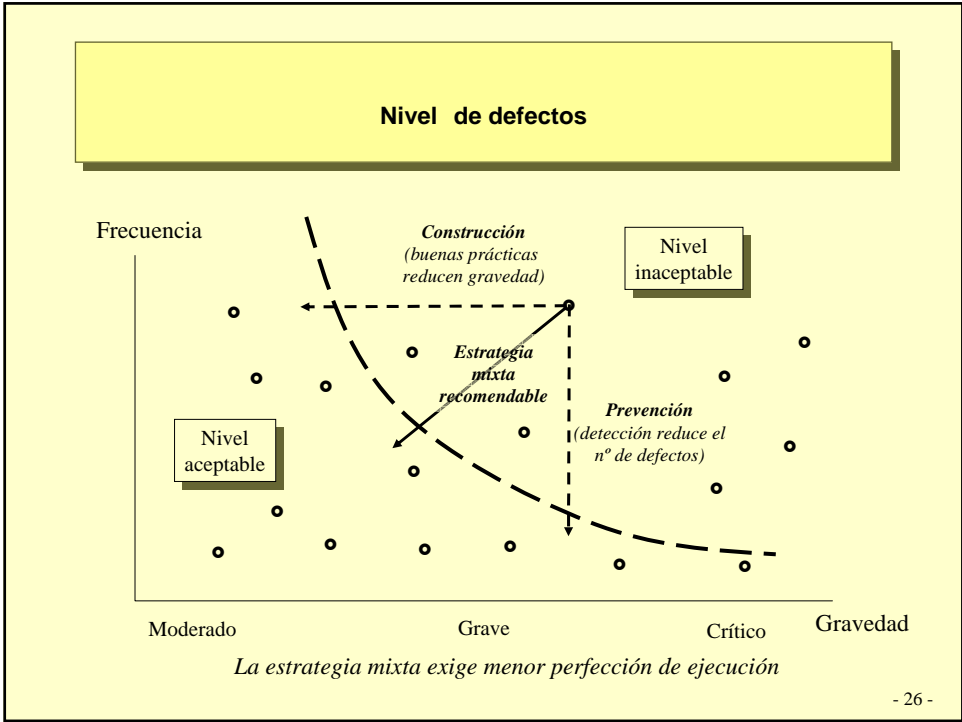
Priorización por riesgos
 Coste = 1 / num. defectos

- 24 -

Verificación y validación



- 25 -



Formación online ATI

• Fo	Conceptos básicos de programación	5
—	Programación Orientada a Objetos	4
—	Bases de Datos(SQL)	4
—	Ingeniería del Software	4
—	Metodologías de Desarrollo	4
—	Java SE	5
	Java JDBC	4
	Programación .NET básica	5
	ADO .NET	4
	Hojas de estilo en cascada CSS	3
	Servidores Web	4
	Java EE	5
	Servicios Web en Java	4
	ASP.NET	4
	Componentes y Servicios Web con .NET	5
	Introducción a la Web, Internet y los Sistemas Distribuidos	3
	HTML	3
	JavaScript básico	3
	XML	4
	Programación .NET básica	5

- 27 -

Formación específica

- Formación mixta: online con sesiones presenciales
 - Requisitos y metodologías
 - Trabajo en equipo, competencias personales
 - Pruebas de software
 - Métricas, Puntos de Función, etc.
 - Procesos, ISO 9001, CMMi, ISO 15504
- También formación in-company y posgrados a través de UAH
- Información a través de calidadsoft@ati.es

- 28 -

Gracias por la atención

Luis Fernández Sanz
Universidad de Alcalá
luis.fernandezs@uah.es

- 29 -